a6 cont

computer-executable procedure to manipulate a global data set, wherein the first server executes the function in response to the receipt of a first work packet, the first work packet containing user context information usable by the first server to perform the sub-task, and wherein the first server transmits the user context information to a second server using a second work packet.

a7

30. The computer-readable medium of claim 29, wherein the work packet contains a reply state, and the computer-readable medium has further computer-executable instructions for: causing the second server to update the work packet by replacing a value contained in the action code with a value contained in the reply state; and causing the second server to send the updated work packet back to the first server.

a8

34. The computer-readable medium of claim 33 having further computer-executable instructions for: in response to receiving a first work packet containing the user context data; causing the first server to determine if a node is in the cache; and if the node is determined not to be in the cache, sending a second work packet containing the user context data from the first server to the second server; causing the second server to retrieve the node from a main memory using the second work packet and store the node in the cache; causing the second server to store a reference to the cached node in the second work packet; and sending the second work packet from the second server to the first server, wherein the first server searches the cached node.

REMARKS

Original claims 1-38 have been examined. No claims have been allowed. Claims 1, 3, 8, 15, 23, 26, 30, and 34 have been amended to correct minor informalities, without narrowing their scope. Applicants appreciate the examiner's time and the courtesy extended during the April 2, 2003 telephonic interview with applicants' representative, Grace Law. The substance of the interview is set forth in the Examiner's Interview Summary Record, which is of record as Paper No. 4. Favorable reconsideration of the claims 1-38 is requested in view of the following remarks.

In the Office action mailed February 12, 2003, claims 1, 6-7, 10-11, 28, 32-33, and 36-37 are rejected for being unpatentable over U.S. Patent No. 6,298,382 to Doi et al. (hereinafter "Doi") in view of U.S. Patent No. 5,452,447 to Nelson (hereinafter "Nelson"). The remaining claims 2-5, 8-9, 12-22, 29-31, 34-35, and 38 are rejected for being unpatentable over Doi in view of Nelson, and further in view of U.S. Patent No. 6,425,021 to Ghodrat (hereinafter "Ghodrat"). Since applicants' arguments are directed to the primary reference of Doi, the §103 rejections have been combined in the following remarks.

APPLICANTS' RESPONSE

Section 103(a) rejection – Claims 1-38

Claims 1-38 are rejected under 35 U.S.C. §103(a) as being unpatentable on the basis of Doi, Nelson, and Ghodrat. The Office action asserts that the "servers" disclosed in the cited reference are the same as the "servers" recited in the claims. This assertion is incorrect because the servers disclosed in Doi are a hardware construct, as opposed to the software construct claimed, and perform vastly different functions from those recited in the claims. Specifically, the "servers" recited in the claims relate to functions for performing tasks within a single procedure (e.g., a computer program) on a computer system, whereas the servers disclosed in Doi are server computers that provide services to client computers on a network. Moreover, in the present application, a computer program is divided into a series of servers (e.g., event loops) defining tasks for the computer program. In contrast, Doi teaches merely the division of data (i.e., a group of documents) being saved onto multiple server computers.

In the present application, a computer program is divided into a series of event loops or "servers," and each server performs a task in the overall computer program (specification, page 12, lines 8-10). In particular, a server defines one or more computer-executable functions for performing a particular task in the overall computer program (Id. at page 6, lines 4-5). For example, a database program could be organized into a server for each of the following functions of the program: buffer management, scanning index pages, scanning data pages, logging, I/O, sorting, updates, and locks (Id. at page 6, lines 5-7). Moreover, each server is sized so that its code and associated data structures fit into the cache 19 of each CPU 21 (Id. at page 13, lines 10-11). The server also has a queue for pending work packets, which contain data needed to perform the task (Id. at page 6, lines 8-12). When a pipelined program is executed on a multiprocessor system, each CPU seeks out a server with a pending work packet, but no two CPUs will be permitted to process a single server's workload unless they can do so without conflict (Id. at page 6, lines 16-17). Under this scheme, each processor will perform a single stage of work in the program for prolonged periods of time, thereby insuring that the global context state for each task tends to remain in the cache of the CPU performing that task (Id. at page 6, lines 2-5). Thus, the data required will be in the cache more often, thereby reducing the number of cache misses in the overall system. The result is increased speed and efficiency of the computer system.

Unlike the servers claimed, the servers disclosed in Doi are hardware constructs. Doi relates to an information retrieval system for retrieving or searching and fetching necessary information from a database (Doi, Col. 1, lines 9-13). In particular, the retrieving system 1 is made up of a master computer 10, a plurality of slave computers 20A to 20B, and a network 30 (Id. FIG. 1, Col. 6, lines 54-57). The slaves 20A to 20B (i.e., search server computers) of Doi refer to stand-alone computers, whereas the servers recited in the claims refer to computer-executable functions for performing sub-tasks of a computer program.

Similarly, the caching file server ("CFS") disclosed in Nelson is not the same as the server recited in the claims. The CFS sets up a common memory storage (i.e., cache) for the file attributes (Nelson, Abstract). The CFS, again, is a computer, and cannot correspond to the server recited in the claims. Thus, contrary to the assertions stated in the Office action, Nelson does not disclose the feature of the first and second servers that are optimized to execute in the cache, as recited in claims 1 and 28.

In regard to Ghodrat, it is not clear from the Office action which of the components of Ghodrat actually correspond to the work packet and the server recited in the claims. In particular, the Direct Memory Access (DMA) and the request/response handlers do not appear to relate to the work packet containing an action code for describing an action to be performed as recited in the claims. Moreover, in Ghodrat, the server 104 is again nothing more than a server computer, and not a "server" as recited in the claims. Thus, contrary to the Office action, Ghodrat does not disclose or suggest the work packet or the server recited in claims 2-5, 8-9, 12-22, 29-31, 34-35, and 38. However, if the Examiner maintains the obviousness rejection in view of Ghodrat, Applicants request that the examiner more clearly specify how the components from the cited reference correspond to the server and the work packet recited in the claims.

As shown, all the cited references refer to server computers, and not "servers" of a computer program, as recited in the claims. Thus, no motivation or suggestion can possibly be drawn from the cited references to make the combination asserted in the Office action. Moreover, even if the cited references are combined as asserted in the Office action, they nevertheless do not disclose or suggest: (1) a server defining a computer-executable function for performing a sub-task of a procedure on a computer system, wherein the servers are optimized to execute in the cache, as recited in independent claims 1 and 28, (2) a server containing instructions for performing a sub-task of a procedure on a computer system having cache, wherein the server is optimized to fit inside the cache when executed, as recited in independent claim 13, and (3) server defines at least one function for performing a sub-task of a computer-executable procedure to manipulate a global data set, as recited in independent claims 23 and 26. Furthermore, in addition to the above-stated reasons, dependent claims 2-12, 14-22, 24-25, 27, and 29-38 are patentable for at least the same reasons as independent claims 1, 13, 23, 26, and 28, from which they respectively depend. Accordingly, applicants request that the Section 103 rejection of claims 1-38 be withdrawn.

In view of the foregoing amendments and remarks, applicants submit that the present application is in condition for allowance.

CONCLUSION

The application is considered in good and proper form for allowance, and the examiner is respectfully requested to pass this application, including pending claims 1-38, to issue. If, in the opinion of the examiner, a telephone conference would expedite the prosecution of the subject application, the examiner is invited to call the undersigned attorney.

Respectfully submitted,

John B. Conklin

John B. Conklin, Reg. No. 30,369
One of the Attorneys for Applicant(s)
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson
Chicago, Illinois 60601-6780
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: May 12, 2003